

Programando um serviço de despertar para seu computador

Tempo de Despertar

Ficar ligando e desligando os computadores que não precisam estar ativos 24 horas por dia pode diminuir sua conta de energia. Neste artigo, vamos dar uma olhada em três diferentes métodos de programar uma hora de despertar no Linux sem necessidade de configurar manualmente a hora na BIOS do computador. **POR MIRKO DÖLLE**



Servidores precisam ficar ligados 24 horas, assim como sistemas RAID. Os usuários costumam deixar suas estações de trabalho rodando ao sair do escritório à noite, embora raramente os computadores sejam necessários por mais de 10 horas por dia. Para aplicações especiais, como servidores de fax que enviam mensagens às 2 da manhã, por que não acordar a máquina logo antes das 2 e colocá-la para dormir novamente durante o dia, até a hora da próxima sessão noturna?

A implementação desses roteiros é bastante simples. Tudo o que você precisa é um temporizador que custe alguns dólares para ligar e desligar as máquinas quando necessário. As coisas começam a ficar caras se, como usuário normal, você não tem acesso ao tempo-

rizador no centro de dados. Nesse caso, normalmente a única solução viável são interruptores de alimentação caros e um controlador computadorizado.

A BIOS da placa-mãe de seu computador tipicamente permitirá que você o desperte sem qualquer hardware extra. Quase qualquer máquina moderna dispõe de uma função para inicialização com hora marcada e não devemos esquecer as funções adicionais “Wake on LAN” e “Wake on Ring”, que permitem que o computador desperte sempre que preciso.

O problema é mudar o horário para o acionamento com hora marcada no Linux – diferentes fabricantes e revisões da BIOS significam a aplicação de métodos diferentes. Por exemplo, o despertador ACPI não funciona em

muitas placas-mãe devido a diferentes implementações do padrão. Por outro lado, o despertador NVRAM pode usar memória não-volátil, ou seja, a memória em que a BIOS armazena seus dados, para permitir edição direta do horário de despertar. Claro, é preciso encontrar a localização correta da memória e sua BIOS deve perceber que os valores mudaram. O terceiro método, *settime*, usa um truque simples. A BIOS sempre desperta na mesma hora e data e o relógio do computador é reconfigurado para uma data específica ao desligar.

ACPI – A Arma Secreta?

O ACPI é provavelmente o modo mais fácil de ligar um computador numa hora pré-definida, contanto que o kernel tenha suporte à placa-mãe que você

escolheu. Se sim, basta armazenar a hora de despertar em `/proc/acpi/alarm`:

```
echo 2004-08-02 20:15:00 >/proc/acpi/alarm
```

O kernel transfere a hora diretamente para o relógio interno (RTC – real time clock) de seu computador, mas não a data. Isso faz com que o computador desperte todos os dias na mesma hora, não apenas na data estabelecida.

O problema é que é um método mais ou menos padronizado de transferir a hora para o RTC, mas não há um padrão desses para a hora de despertar. Conseqüentemente, a função `acpi_system_write_alarm` em `drivers/acpi/system.c` pertencente ao kernel 2.4 ou ao `drivers/acpi/sleep/proc.c` do kernel 2.6 apenas dá suporte aos campos do horário de alarme.

Sem Data

Os desenvolvedores do kernel Andy Grover e Paul Diefenbaugh já prepararam o caminho para a funcionalidade plena, mas necessitam de uma tabela ACPI fixa utilizável (FACP). Se você for conferir os comentários no código do kernel, notará que as tabelas FACP atuais para placas-mãe são inutilizáveis, o que fez com que o bloco de código fosse desabilitado.

Algumas placas-mãe não repassam a hora diretamente ao relógio do computador, armazenando a hora do último boot em vez da hora de despertar desejada. Outras travam imediatamente quando se faz uma tentativa de escrever em `/proc/acpi/alarm`.

O despertador ACPI funciona com muitas das placas-mãe desta

safrá. Dê uma olhada na seção de distribuição na homepage do LinVDR (Linux Video Recorder) para ver uma pequena lista dos modelos suportados.[1]

Trapalhão

Se você quer usar o ACPI para despertar seu computador, precisa certificar-se que as configurações da BIOS dele estão corretas. Há anos os computadores têm funções da BIOS chamadas *Wake on Timer*, *Resume on Alarm*, *RTC Alarm Resume* ou variações similares sobre o mesmo tema. Dito isso, esperava-se que os usuários configurassem a hora na BIOS. O despertador ACPI usa uma técnica similar, mas não usa a função BIOS. É por isso que você precisa desabilitar a função de despertar da BIOS na maior parte das placas-mãe. Note também que a hora marcada para a próxima inicialização não é mostrada na BIOS.

Outro obstáculo: há algumas placas-mãe, como a Asus A7V133, que não aceitam de bom grado um script de desligamento que use o `hwclock -w` para sincronizar o relógio interno com a hora do sistema depois que a hora de

despertar é informada. O computador simplesmente não desperta novamente. É preciso configurar a opção `--directisa` ao chamar o `hwclock` ou, alternativamente, configurar a hora de despertar após chamar o `hwclock`.

Feitiço do tempo

Como já mencionamos, o despertador ACPI diz ao computador que desperte todos os dias na hora pré-configurada; isso tende a restringir a utilidade da função. O despertador ACPI seria muito útil para um servidor de fax que manda mensagens automaticamente todas as noites antes de ser novamente desligado. Ele também seria bastante útil para gravadores de vídeo no PC (PVR – *Personal Video Recorder*), por exemplo o VDR. A maior parte dos canais atualiza seus guias eletrônicos de programação (Electronic Program Guide - EPG) entre a meia-noite e as duas da manhã, de forma que ligar a máquina às duas da manhã permitiria sincronizar o EPG com a programação de filmes do usuário – para gravá-los, por exemplo.

Porém, um servidor de impressão que seja ligado às 9 da manhã e desli-

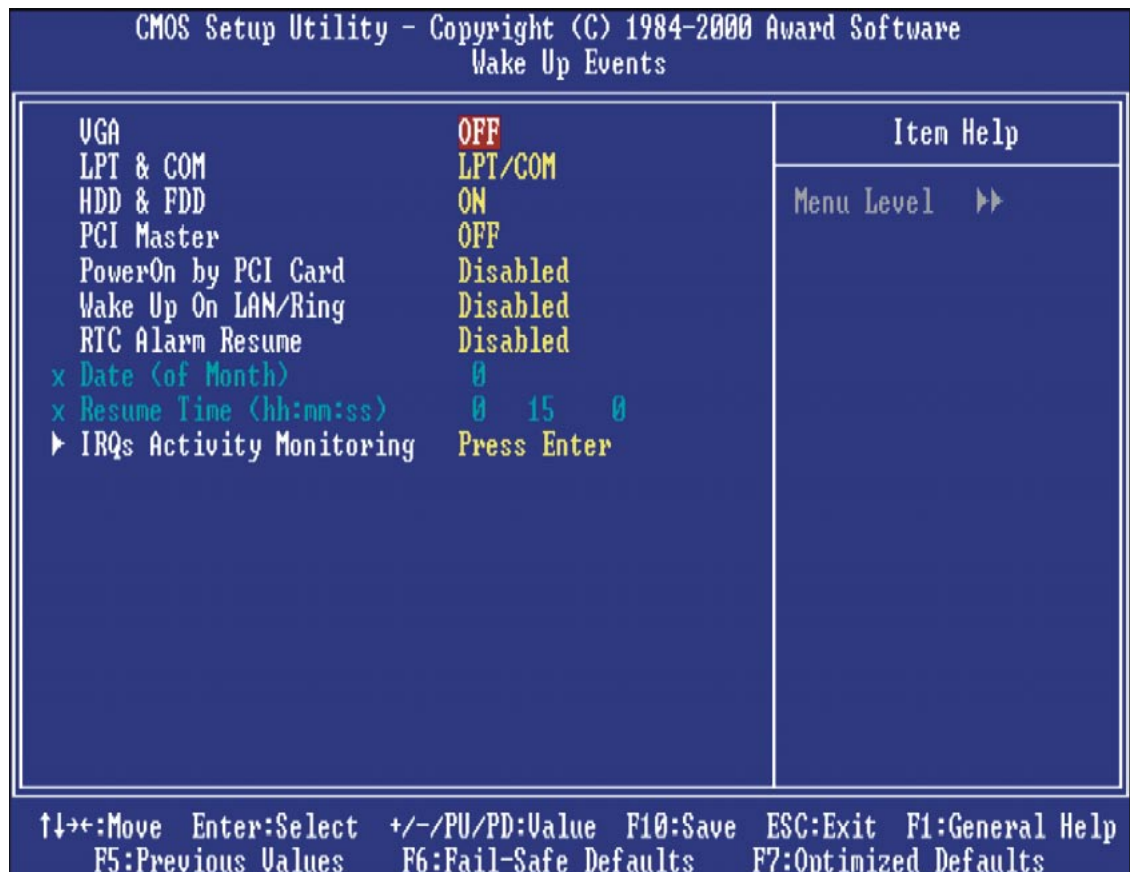


Figura 1: A maioria das BIOS de PCs têm uma função para despertar a máquina chamada *RTC Alarm Resume* ou algo similar.

```

bash
dovei:~# nvram-wakeup -N -s `date -d "2004-08-15 22:20:00" +%s` -n -C /
~wakeup.conf
All values are displayed as they are stored in the nvram/rtc.
(And do not correspond necessarily to the system date/time.)
WakeUp : Enabled (0xFF)
Day      : 15 (0x0F)
Hour     : 20 (0x14)
Minute  : 15 (0x0F)
Second  : 00 (0x00)
Enabling (0xFF) WakeUp-on-RTC in nvram.
New Day  : 15 (0x0F)
New Hour : 20 (0x14)
New Minute : 15 (0x0F)

```

Figura 2: Para um teste bem-sucedido, os valores exibidos pelo `nvram-wakeup` precisam combinar. A hora de despertar configurada na BIOS é mostrada no alto; o número de baixo mostra o valor analisado pelo programa.

gue às 7 da noite estaria ligado e funcional por 10 horas também nos fins de semana. Uma rotina adicional nos scripts de inicialização que reconheça fins de semana e feriados poderia desligar a máquina, mas essa não passa de uma solução tapa-buraco que também tem seus problemas: se você realmente precisasse usar o servidor de impressão num fim de semana ele se desligaria sozinho mesmo que fosse inicializado manualmente, o que é irritante.

Um script que busque fins de semana e feriados ao desligar o computador e programe o dia e a hora de despertar seria uma solução muito melhor. O despertador NVRAM e o método `settime` têm a resposta.

Despertador NVRAM

O despertador NVRAM usa as configurações da BIOS armazenadas na RAM não-volátil (NVRAM – *Non Volatile RAM*). O módulo `nvram` do kernel permite que o Linux acesse um máximo de 128 bytes de memória não-volátil.

Para que isso funcione, é necessário compilar o módulo `nvram` do kernel e também criar os dispositivos de caractere `/dev/nvram` com o número `major` com valor 10 e o número `minor` com valor 144, `/dev/rtc` com número `major` em 10 e `minor` em 135 e `/dev/mem` com `major` em 1 e `minor` em 1. Além disso, é preciso modificar a configuração do módulo no arquivo `/etc/modules.conf` para mandar o computador carregar o módulo NVRAM quando o dispositivo `/dev/nvram` for acessado ou imediatamente após a inicialização.

Usando as funções da BIOS

Não há armadilhas ao se compilar o programa a partir do código fonte (disponível em [2]). O caminho padrão de

instalação é `/usr/local`, mas os administradores podem editar o `Makefile` para mudar o caminho para `/usr`, se necessário. Se os dispositivos `/dev/nvram`, `/dev/rtc` e `/dev/mem` não existirem ainda, use o comando `make devices` para criá-los automaticamente.

Na versão 0.96 os comandos `tr` nas linhas 62 e 75 do script auxiliar `guess-helper.sh` não funcionam:

```

answers=`$echo $answers |
tr [:lower:] [:upper:]`

```

Felizmente, o script funcionou bem após acrescentar quatro apóstrofes.

```

answers=`$echo $answers |
tr '[:lower:]' '[:upper:]'`

```

Para a maioria dos fabricantes, o modo como o NVRAM é alocado está envolto numa espessa neblina de segredos – você pode ter alguma dificuldade para encontrar o lugar em que a data e a hora foram armazenadas, pela falta de documentação. O pior é que a localização das memórias tende a ser um alvo móvel, que muda a cada vez em que a BIOS é atualizada.

Os arquivos fonte do despertador NVRAM incluem especificações para algumas placas. Esses detalhes são cortesia do script `guess-helper.sh`. Rodar o `nvram-wakeup` com a opção `-D` inicia o programa em modo debug e diz se a placa mãe é conhecida. Porém, se o comando `nvram-wakeup` diz que:

```

nvram-wakeup: Your motherboard
is currently not supported.
(Sua placa-mãe não é suportada)

```

será preciso rodar o script `guess-helper.sh`.

O script `guess-helper.sh` tenta localizar a posição dos campos de memória NVRAM. Você precisa reinicializar seu computador ao menos quatro vezes para fazê-lo. Na primeira vez, é preciso desabilitar a função de despertar da BIOS, ajustar a data para o dia 31 do mês e a hora para um segundo antes da meia-noite. Em seguida, deve-se chamar `guess-helper.sh`. Ajuste então a hora de despertar para o dia 11 do mês, às 12:13:14, depois para o dia 1º, meia-noite, e finalmente desabilite novamente a função de despertar. A cada vez que o ponto de despertar é chamado, `guess-helper.sh` compara o conteúdo da NVRAM com as outras chamadas para estabelecer a localização dos pontos de despertar na memória. O `guess-helper.sh` usa dois métodos para fazê-lo: acesso via `/dev/nvram` e acesso direto ao endereço I/O. Isso resulta em dois arquivos de configuração, `nvram-wakeup.conf` no diretório `guess-nvram-module` e `guess-directisa` no diretório pessoal do `root`.

Posicionamento da Informação

Uma olhada rápida nos arquivos de configuração nos mostra onde estão armazenados os comentários sobre a placa-mãe e informações sobre ende-

Listagem 1: `nvram-wakeup.conf`

```

#####
## motherboard autodetection information:
##
## - motherboard vendor: ""
## - motherboard type: "VT8367-8235"
## - motherboard revision: ""
## - BIOS vendor: "Award Software International, Inc."
## - BIOS version: "6.00 PG"
## - BIOS release: "04/07/2003"
addr_stat = 0xD2
shift_stat = 5
addr_day = 0xD8
addr_hour = 0xD9
addr_min = 0xDA
addr_sec = 0xDB
upper_method = VT8235_37

```


reços. No caso da placa Elite Epox 8K5A2+, o arquivo `/root/guess-nvram-module` estava vazio, exceto por alguns comentários, enquanto o `/root/guess-directisa` continha os endereços mostrados na Listagem 1.

Se ambos os arquivos de configuração estiverem vazios, podemos considerar que `guess-helper.sh` foi incapaz de localizar os endereços.

É preciso dizer ao programa qual o caminho para o arquivo de configuração, usando o parâmetro `-C`, sempre que você chamar `nvram-wakeup`. Além disso, o parâmetro `-A` é requerido nesse caso, já que estaremos usando os endereços de I/O para acesso direto, em vez do dispositivo `/dev/nvram`. Ambos os parâmetros podem ser desconsiderados se sua placa-mãe for suportada pelo `nvram-wakeup`.

Testando

Para nossa rodada de testes com o `nvram-wakeup`, precisamos de uma hora arbitrária de despertar na BIOS; usaremos o `nvram-wakeup` para validar os resultados. No exemplo, a BIOS foi ajustada para despertar o computador às 20:15h do dia 15. A data para a chamada do `nvram-wakeup` precisa considerar também o vetor de tempo entre a hora local e o UTC e cerca de cinco minutos para que o computador inicialize. A Figura 2 mostra a saída do `nvram-wakeup`.

A seção superior da Figura 2 mostra os dados de despertar que o `nvram-wakeup` localizou na NVRAM de sua máquina; o valor de baixo mostra os dados lançados, sem o parâmetro de escrita protegida, `-N`. Os valores combinam, de forma que podemos constatar que o `guess-helper.sh` realmente localizou as posições de memória corretas. Em seguida é preciso configurar uma data cerca de 15 minutos no futuro e então desligar o computador:

```
nvram-wakeup -s `date -d "+15`
Minutes" +%s` -A -C /etc/nvram`
-wakeup.conf
poweroff
```

Se o computador não despertar como esperado, isso pode ser devido a um problema básico com APM/ACPI (veja artigo na página 64) ou talvez seja pre-

ciso rodar a rotina da BIOS novamente – ou seja, reiniciar o computador – para despertar na hora pré-configurada.

Reiniciar, e não Desligar

Há um modo muito fácil de descobrir se você está enfrentando o chamado problema de reboot. Configure uma hora de despertar para os próximos quinze minutos e digite `reboot` para reiniciar sua máquina. Ajuste então uma hora de despertar para daí a quinze minutos e digite `poweroff`.

Se a BIOS despertar nos quinze minutos seguintes – ao contrário do que ocorreu em nosso teste anterior – seu computador tem que ler o conteúdo da BIOS para que o `wakeup` funcione.

Para algumas placas-mãe, o tamanho da área da memória acessível ao dispositivo `/dev/nvram` como definido no módulo `nvram` é pequeno demais para acessar a área inteira do NVRAM. Os casos em que o `guess-helper.sh` é incapaz de localizar as posições de dia, hora, minuto e segundo ou apenas acha alguns campos são indicativos desse problema. Os arquivos fonte do kernel definem a área em `drivers/char/nvram.c`:

```
#define NVRAM_BYTES 2
(128-NVRAM_FIRST_BYTE)
```

É preciso estender a área para 128 bytes completos:

```
#define NVRAM_BYTES 128
```

Essa mudança pode ajudar o `guess-helper.sh` a localizar os campos.

De Volta para o Passado

`settime` é um método engenhoso, de funcionamento garantido em qualquer

placa-mãe. A idéia é programar uma hora fixa para despertar na BIOS, por exemplo o dia 31 de agosto às 23:59:59. Ao desligar sua máquina, um script calcula o vetor de tempo entre o desligamento e o despertar, subtrai esse vetor de 31 de agosto de 2004, às 23:59:59, e ajusta o sistema e a hora do relógio interno para um dia e hora apropriados em agosto de 2004. Da próxima vez que iniciar seu computador, terá apenas de corrigir a configuração da hora.

Em aplicações práticas, esse método pode ser um tanto incômodo. É preciso ter certeza de haver configurado a hora certa todas as vezes em que a máquina for ligada e antes que o `fsck` seja executado – caso contrário, isso pode evitar que as verificações de disco sejam completadas. Se você corrigir a hora antes que o `fsck` seja executado, não há lugar para anotar o fato, já que você não terá acesso de escrita nas partições. Sistemas com dual boot com Windows ou alguns outros sistemas operacionais também são manhosos, pois despertariam no dia e na hora errados. Além disso, esse método só faz sentido se você reiniciar a máquina ao menos a cada dois meses.

Delta T

Dois scripts separados trazem a resposta: o `settime` calcula o vetor de tempo entre a hora real e a de despertar, armazena a diferença no arquivo `/etc/timediff`, configura a hora do sistema para uma data em agosto de 2004 e sincroniza o RTC [3]. Ao inicializar o sistema, `correcttime` é chamado no primeiro script de inicialização, lê o vetor de tempo, corrige a hora do sistema e sincroniza o RTC. Isso assegura que os outros scripts de inicialização nem mesmo notarão a alteração de horário.

Listagem 2: Script `settime`

```
#!/bin/bash
BiosWakeup="2004-08-31 23:59:59"
Wakeup=`date -d "$1" +%s`
Now=`date +%s`
Bios=`date -u -d "${BiosWakeup}" +%s`
Diff=$((Wakeup)-$Now)
echo "$[${Now}-$Bios]}" > /etc/timediff
date -u -s "${BiosWakeup} ${Diff} seconds ago" >/dev/null
hwclock -w --utc
```

O método *settime* baseia-se no fato de que o relógio interno continua a manter a hora normal. Assim, não há mais necessidade de sincronizar a hora do sistema com outro sistema ou usar um relógio exato sempre que o computador for religado. Tudo o que você precisa saber é o número de segundos que o relógio perdeu em relação à hora real.

Além disso, não há necessidade de realizar cálculos complicados entre datas, incluindo anos bissextos; o utilitário *date* tem todas as funções necessárias. Por exemplo:

```
date -s "+3600 seconds"
```

adianta em uma hora o relógio do sistema. Para voltar ao passado, tente o seguinte comando:

```
date -s "3600 seconds ago"
```

Além disso, *date* pode converter datas para a notação internacional padrão, segundos desde o *epoch*, a data do “big bang” do Unix (zero horas do dia 1º de janeiro de 1970):

```
debian:~# date -d "2004-08-31 23:59:59" +%s
1093993199
```

O resto é fácil. A Listagem 2 mostra um trecho do script *settime*. A Linha 3 converte a hora de despertar para segundos após o epoch, a linha 4 estabelece a data atual em segundos após o epoch e a linha 6 usa esses valores para calcular o vetor de tempo em segundos.

A linha 5 calcula a hora de despertar configurada na BIOS, um segundo antes da meia-noite de 31 de agosto de 2004 neste exemplo, em segundos após o epoch, permitindo que a linha 7 calcule a diferença para a hora real e armazene o valor no arquivo */etc/timediff*. Agora, na próxima vez em que o sistema inicializar, o *correcttime* precisa apenas acrescentar a diferença à hora do sistema para obter a hora real.

Finalmente, as linhas 8 e 9 ajustam a hora do sistema e do relógio interno para um valor no passado. Como já foi visto na linha 5 do script, essas horas são convertidas para UTC e restauradas. Para usar a hora local em vez da UTC para o relógio interno, é preciso mudar

as três linhas para que reflitam seu fuso horário local.

Corrigindo a Hora na Inicialização

O script *settime* só é chamado quando necessário, ou seja, quando é preciso despertar a máquina num horário pré-estabelecido e imediatamente antes de desligar. Isso significa que o *correcttime* precisa descobrir se a hora do relógio interno é a hora real ou se o computador está em algum lugar do passado. A Listagem 3 mostra um trecho do *correcttime*.

A chamada a *date* na linha 3 descobre quando o */etc/timediff* foi modificado pela última vez, devolvendo um valor em segundos após o epoch. Se a data de modificação na linha 5 estiver no futuro, a hora do sistema deve ter sido configurada para uma data no passado. Se a data de modificação estiver no passado, a hora do sistema deve ser a hora real atual.

Das linhas 6 até 8, o *correcttime* lê o vetor de tempo entre o sistema e o tempo real do arquivo */etc/timediff* e o acrescenta à hora atual do sistema, levando assim a máquina de volta à hora real. A hora real é transferida então para o relógio interno, já que a maioria das distribuições não sincroniza explicitamente a hora do sistema com o valor do relógio interno durante o boot.

Poupe energia, poupe dinheiro!

Uma olhadela em sua última conta de luz mostra o quanto a eletricidade é cara. Você pode poupar energia – e sem sofrer efeitos colaterais, se você usar um pouco de cérebro no processo. Cinco PCs desktop normais (150 watts cada, no mínimo) ligados noite e dia, como é

o caso em muitos escritórios, consomem mais de 6500 quilowatts hora, ou R\$ 3850,00 (aproximadamente US\$1500.00 ou 1200 Euros) por ano. Os servidores são ainda piores, pois valores de 400 watts ou mais por unidade são bastante comuns. Um único servidor pode consumir mais de 3500 quilowatts hora, o que acrescentaria R\$ 2000,00 (US\$780.00 ou 630 Euro) à sua conta de luz anual.

Desligar os computadores automaticamente às 9 da noite e ligá-los novamente às 6 da manhã seguinte significaria uma economia de mais de 3600 kWh para os desktops e cerca de 2000 kWh para o servidor.

Isso é equivalente a uma economia de quase R\$ 3200,00 (US\$ 1200.00 ou 1000 Euros), para não mencionar a redução nas emissões de CO₂ e outros ganhos ambientais.

Se você precisa dos computadores além do horário normal do escritório, basta apertar o botão. O servidor responderá a “Wake on LAN” a partir do script inicial da estação de trabalho, por exemplo. ■

Listagem 3: Script *correcttime*

```
#!/bin/bash
if [ -r /etc/timediff ]; then
    Timediff=`date -r /etc/timediff +%s`
    Now=`date +%s`
    if [ "${Timediff}" -gt "${Now}" ]; then
        Diff=`cat /etc/timediff | head -n 1`
        date -s "+${Diff} seconds" >/dev/null
        hwclock -w --noadjfile --utc
        exit 0
    fi
fi
```

INFORMAÇÕES

[1] Compatibilidade do ACPI:
<http://linvdr.org/wiki/index.php?pagename=LinVDR-Mainboards>

[2] Despertador NVRAM:
<http://sourceforge.net/projects/nvram-wakeup/>

[3] Código-fonte dos scripts *settime* e *correcttime* no site da Linux Magazine:
<http://www.linux-magazin.de/Service/Listings/2004/08/wakeup>

[4] ACPI: <http://www.acpi.info>

[5] ACPI na Wikipedia:
<http://en.wikipedia.org/wiki/ACPI>